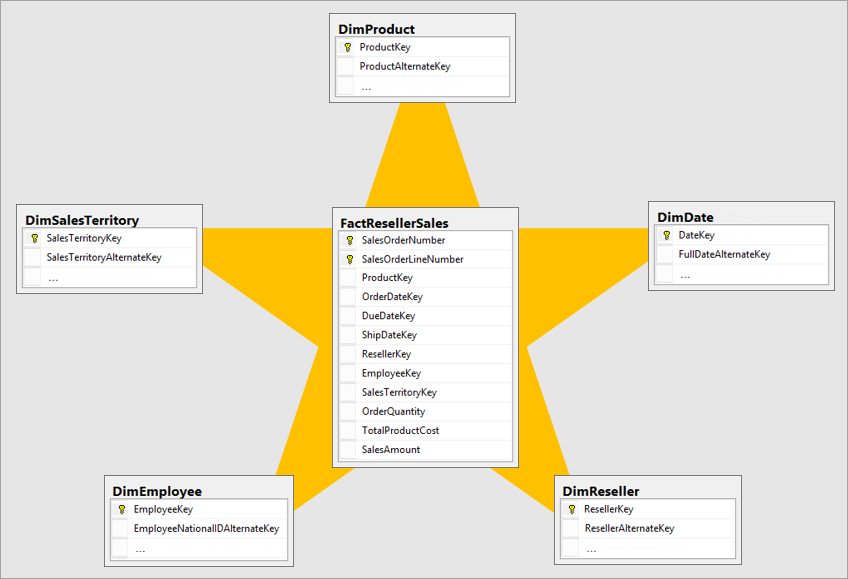
We will convert the Postgresql-database-diagram to STAR SCHEMA DIAGRAM

Example:



The facts tables are usually those tables around which the analysis is carried on.(In our case its sales/payment/revenue.)

Refer 01.Step document for the analysis topic.

Now there are multiple tables spread across the diagram we can combined few of the column from different tables to one table that gives similar information or that shares similar data.

Star Schema has been modeled in draw.io.

Now create the tables in database.

1.Open query tool on right clicking dvdrental database.

CREATE TABLE dimdate

(

date\_key integer NOT NULL PRIMARY KEY,

date date NOT NULL,

year smallint NOT NULL,

quarter smallint NOT NULL,

month smallint NOT NULL,

day smallint NOT NULL,

week smallint NOT NULL,

is weekend boolean

);

Now you can check this table under Tables.

To check the information about dimDate table

select from information\_schema.columns where table\_name = 'dimdate'

If you do not want all the information than you can select the specific info like below

select column\_name, data\_type from information\_schema.columns where table\_name = 'dimdate'

CREATE TABLE dimcustomer

(

customer\_key SERIAL PRIMARY KEY,

customer\_id smallint NOT NULL,

first\_name varchar(45) NOT NULL,

last\_name varchar (45) NOT NULL,

email varchar (50),

address varchar (50) NOT NULL,

address2 varchar (50),

district varchar (20) NOT NULL,

city varchar (50) NOT NULL,

country varchar (50) NOT NULL,

postal\_code varchar(10),

phone varchar (20) NOT NULL,

start\_date date NOT NULL,

end\_date date NOT NULL

);

CREATE TABLE dimmovie

(

movie\_key integer NOT NULL PRIMARY KEY,

film\_id integer NOT NULL,

title varchar (255) NOT NULL,

description varchar (200) NOT NULL,

release\_year integer NOT NULL,

language\_id smallint NOT NULL,

length smallint NOT NULL,

ratings varchar (10) NOT NULL,

special\_features varchar (100) NOT NULL,

);

CREATE TABLE dimstore

(

store\_key integer PRIMARY KEY,

store\_id integer NOT NULL,

address varchar(50) NOT NULL,

address2 varchar (45),

district varchar (50),

city varchar (50) NOT NULL,

country varchar (50) NOT NULL,

postal\_code varchar(10),

manager\_first\_name varchar (45) NOT NULL,

manager\_last\_name varchar (45) NOT NULL,

start\_date date NOT NULL,

end\_date date NOT NULL

);

CREATE TABLE factSales

(

sales\_key SERIAL PRIMARY KEY,

date\_key integer REFERENCES dimDate (date\_key),

customer\_key integer REFERENCES dimCustomer (customer\_key),

movie\_key integer REFERENCES dimMovie (movie\_key),

store\_key integer REFERENCES dimStore (store\_key),

sales\_amount numeric

);

Insert data into the above tables.

INSERT INTO dimdate

(date\_key, date, year, quarter, month, day, week, is\_weekend)

SELECT

DISTINCT (TO\_CHAR (payment\_date :: DATE, 'yyyMMDD') ::integer) as date\_key, #the payment date is in timestamp format so to convert it to integer.

date (payment\_date) as date,

EXTRACT (year from payment\_date) as year,

EXTRACT (quarter FROM payment\_date) AS quarter,

EXTRACT (month FROM payment\_date) AS month,

EXTRACT (day FROM payment\_date) AS day,

EXTRACT (week FROM payment\_date) AS week,

CASE WHEN EXTRACT (ISODOW FROM payment\_date) IN (6, 7) THEN true ELSE false END AS is\_weekend #to select the days which is weekend ..Monday 1….Sunday 7

FROM payment;

INSERT INTO dimcustomer(customer\_key,customer\_id,first\_name,last\_name,email

,address,address2,district,city,country,postal\_code,phone,start\_date,end\_date)

SELECT c.customer\_id as customer\_key,

c.customer\_id,

c.first\_name,

c.last\_name,

c.email,

a.address,

a.address2,

a.district,

ci.city,

co.country,

a.postal\_code,

a.phone,

now() AS start\_date,

now() AS end\_date

FROM customer c

JOIN address a ON (c.address\_id = a.address\_id)

JOIN city ci ON (a.city\_id = ci.city\_id)

JOIN country co ON (ci.country\_id = co.country\_id);

INSERT INTO dimmovie(movie\_key,film\_id,title,description,release\_year,language\_id,length,ratings, special\_features)

SELECT f.film\_id as movie\_key,

f.film\_id,

f.title,

f.description,

f.release\_year,

f.language\_id,

f.length,

f.rating,

f.special\_features

FROM film f;

INSERT INTO dimstore(store\_key,store\_id,address,address2,district,city,country,postal\_code,manager\_first\_name,manager\_last\_name,start\_date,end\_date)

SELECT s.store\_id as store\_key,

s.store\_id,

a.address,

a.address2,

a.district,

ci.city,

co.country,

a.postal\_code,

st.first\_name as manager\_first\_name,

st.last\_name as manager\_last\_name,

now() AS start\_date,

now() AS end\_date

FROM store s

JOIN staff st ON (s.manager\_staff\_id = st.staff\_id)

JOIN address a ON (s.address\_id = a.address\_id)

JOIN city ci ON (a.city\_id = ci.city\_id)

JOIN country co ON (ci.country\_id = co.country\_id);

INSERT INTO factSales (date\_key, customer\_key, movie\_key, store\_key, sales\_amount)

SELECT

DISTINCT (TO\_CHAR (payment\_date :: DATE, 'yyyMMDD') ::integer) as date\_key,

p.customer\_id as customer\_key,

i.film\_id as movie\_key,

i.store\_id as store\_key,

p.amount as sales\_amount

FROM payment p

JOIN rental r ON (p.rental\_id = r.rental\_id)

JOIN inventory i ON (r.inventory\_id = i.inventory\_id);

Query to see the data

SELECT f.title, EXTRACT (month FROM p.payment\_date) as month, ci.city, sum (p.amount) as

FROM payment p

JOIN rental r ON (p.rental\_id = r.rental\_id)

JOIN inventory i ON (r.inventory\_id = i.inventory\_id)

JOIN film f ON (i.film\_id = f.film\_id)

JOIN customer c ON ( p.customer\_id = c.customer\_id)

JOIN address a ON (c.address\_id = a.address\_id )

JOIN city ci ON ( a.city\_id = ci.city\_id)

group by (f.title, month, ci.city)

order by f.title, month, ci.city, revenue desc;

similarly with other tables we can use the above query.Difference between this stars schema model queries and 3NF model query(query which we used earlier in previous document) is that the star schema runs faster.